Choosit/User Defined Mapping

ENABLING USERS TO CONFIGURE A PROCESS OF MAPPING SECTIONS OF BROWSER-ACCESSIBLE CONTENT

INV: Dave Morton
David Lloyd Summers

Summary

User Defined Mapping (UDM) provides a method of accessing browser-based data or information, such as Internet content, networked applications, custom data, field input and passowords, from the UDM Choosit browser. Choosit is the tool used to map specific, user-defined content, such as text or text links, and makes it available on any communications device, such as a rotary or cellular telephone, mobile PDA, or PC. Choosit allows users to voice enable specific sections of networked data by speaking a name assigned by the user.

I. Background

Problem

The basic problem arises when a user wants to access browser-based data on his/her device using a voice interface. However, this cannot be done by existing methods because most of the content currently consists of formats such as images with JavaScript. This hinders the ability to access information over a voice interface in the same way a person might access networked content via traditional access methods, such as PC-based Internet browsing. For example, if a user goes to a Web site, s/he can navigate to a section within that Web site that contains articles that s/he wants to access. However, within those articles are numerous others. If the user accessed those articles using a voice browser, the voice browser would start reading every link and would finally gets to the stories the user actually wanted after the system had taken up a significant amount of time reading the smaller links.

Talk2's Choosit Solution

The traditional PC-based user experience of accessing data through a browser cannot be exactly recreated over the phone so the next best alternative is to enable the user to decide what gets read to him/her on the phone. Choosit is able to create a map to the location or section (not just static text) that the user selects and wants to access, via such methods as voice recognition or Internet-enabled devices. The difference between the section and the static text is important and a key component of UDM because the section is dynamically updated every time a user go to the site, the text is not. This allows the text to change as content is updated because Choosit has enabled the user to select the section or position in which the text would appear.

While UDM also has the capability of mapping static text in addition to dynamic sections of data, it is desirable for the user to map all sections s/he would like to access through a device because if content has any formatting at all, a device is not going to read the formatted text very well. One example might be a stock portfolio on a Web site that a user would like read to him/her. Typically, the information is in a table and the end user wants the system to read the name of the stock and its value. The table may contain other information that the user does not really want, so if the user sets the site up so Choosit will select the name and value, the system will actually offer only the information that had been mapped. In addition, UDM will enable the mapped information – in this case, the value of the stock – to be updated dynamically so the user always receives the most current value.

UDM also differs from other voice-enabling or text-to-speech solutions such as VoiceXML. Other solutions require the person who designed the site to go back and redesign the site to be compatible with

that solution. However, this site designer may not care about the end user's needs in every case. With UDM technology the end user actually does the mapping, which offers the user two advantages. First, it does not matter what the owner of the content wants the end user to access, the user can just map the content or data and the owner of the data does not need to be involved. In this case of Choosit technology, the user controls content similar to way it is possible to disable formatting such as images on the browser. For example, if a content provider writes in VoiceXML, s/he can push content that s/he wants the end user to hear or see. The second advantage is that the end user has the ability to choose what s/he wants to hear or see.

In addition to giving the end user the ability to map desired content for access over a voice interface, UDM also has the ability for others, besides simply the end user, to map data. For example, UDM allows a system administrator to map data behind a corporate firewall, such as a corporate intranet or CRM application, for remote access.

II. How UDM Works – A High Level Overview

1. The user navigates to the Choosit Web page from the primary entry portal.
2. The navigational tool loads in the left frame. On the right side of the page is a "working desktop" where the site of interest is loaded. (See Figure 1)
3. The user is prompted to enter the desired content, such as a URL or custom application, which will fill the working desktop on the right side of the screen. The content on the working desktop will look like the original content form, although Talk2 will have stripped out all extraneous formatting, such as the JavaScript, logos, images, etc. Essentially, Talk2 will have remapped all aspects of the content, including such items as forms or images, so that the content will work with "Choosit." (See Figure 2 for a complete map of the Choosit end user experience)
4. The user is prompted to choose a name for the content, which acts as a bookmark or main folder. This "bookmark" is any user-chosen name or description given to the desired data that the user will access. For example, "Sports" may be a bookmark name for a sports-related URL the user has chosen.
5. After Choosit has loaded the page, it asks, "Is there anything here you would like to map?" If the user says "yes," then Choosit says "text" or "link," and the user clicks on the mouse to drag and select the text that s/he wants from the links. The user selects the desired text or link and is asked to select a "voicelink" name that will be spoken over the telephone to access the content. Choosit would say, "Do you want to add text or links?" If the user selects "text" then no matter what is selected, the content is going to be accessed by the user on the device as text. Often, the user may want links read over the voice portal – for example, temporal links may be the headlines of a story. The user selects this option by clicking on the mouse and holding it down to highlight the desired section. The highlighting will follow the format rather than the shape of the text (for example, a rectangular region).
6. When the user releases the mouse, Choosit will ask, "Do you want to add another text area?" The user can keep adding links or text by repeating the process. When the user is finished selecting a section, Choosit will say, "Enter a name," and the user enters the name of the voicelink. The link is loaded, assigned the name, and placed in the main folder (or bookmark). Once you select a name for the voicelink, a branching tree is formed with the original bookmark link. A bookmark may have several links under it. When a user selects the bookmark from his/her device, Choosit will enable the system to say for example, "This bookmark has the following links, A, B, C."
7. Once the user selects the voicelink within the bookmark, the system would begin offering text at that point. If the voicelink contains additional links, the system will offer the links to the user so that s/he can choose among the options. For example, if a voicelink is a page of a Web site and contains a collection of articles, the system would list the article titles so the end user could select among the options.
8. Once the voicelink is established, the user can start over again and add more voicelinks or links within the voicelinks. The system will ask if there's anything the user would like to map and if the user says

"no," then the links all just become active, the user can click on a link and follow that link, just like a normal browser. The user can go even deeper before actually mapping the content.

Figure 1 illustrates the graphic user interface for UDM. There are three sections on the left and on the top there is the Choosit logo.

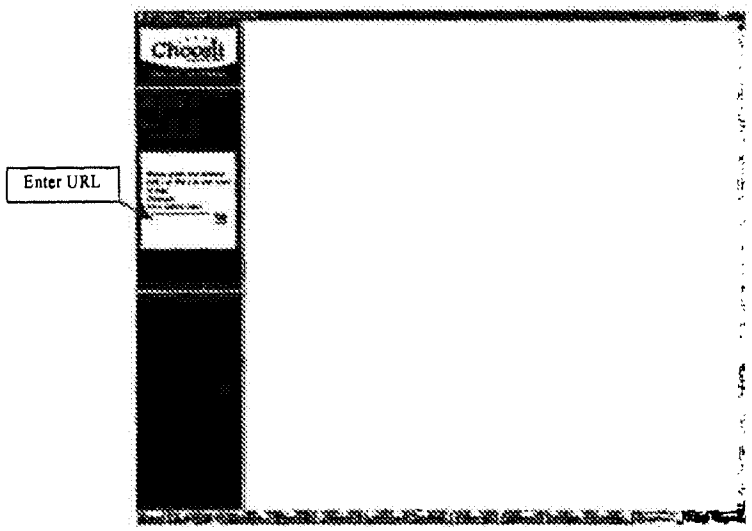


Figure 1

Call Flow Explanation Figure 2

UDM Prompt	UDM output	
	Left frame	**Right frame**
Please enter the address (URL) of the site you want to map	1. Entered URL name output below "Voice Enabling"	
Enter a name for this bookmark. This will be the name you speak to access this content.	1. Entered bookmark name output below "Book Mark Name:" 2. If the required page is successful loaded, "Is there any information here you would like to map?" otherwise "Please enter the address (URL) of the site you want to map" is output.	If the required page is successful loaded, page information is displayed; otherwise standard error message is output.
Is there any information here you would like to map?	Click "Yes", "Do you want to add text or links from this page?" is output.	

	Click "No", "Using the screen on the right follow the link to the page you want to map." is output.	
Do you want to add text or links from this page?	Click "Text", "On the screen at the right, select an area of text by clicking and dragging over the select text." is output.	
	Click "Links", "On the screen at the right, select a link or group of links by clicking and dragging your mouse over the group." is output.	
Using the screen on the right follow the link to the page you want to map.	Click the selected link, If the required page is successful loaded, "Is there any information here you would like to map?" is output; otherwise "Please enter the address (URL) of the site you want to map" is output.	If the required page is successful loaded, page information is output; otherwise standard error message is output.
On the screen at the right, select an area of text by clicking and dragging over the select text.	Drag the select text, "Do you want to add another text area to this voicelink?" is output.	
On the screen at the right, select a link or group of links by clicking and dragging your mouse over the group.	Drag the select links, "Are the pages these links lead to similar in layout?" is output	
Do you want to add another text area to this voicelink?	Click "Yes", "On the screen at the right, select an area of text by clicking and dragging over the select text." is output.	
	Click "No", "Enter a name for this voicelink." is output.	
Are the pages these links lead to similar in layout?	Click "Yes", "On the screen at the right, select an area of text by clicking and dragging over the select text." is output if load is successful; otherwise "Please enter the address (URL) of the site you want to map" is output.	If the first select link is successful loaded, the page information is output; otherwise standard error message is output.
	Click "No", "Enter a name for this voicelink." is output.	
Enter a name for this voicelink.	Entered name is added into bookmark tree. Click "Ok" after enter the name of the voicelink, "Do you want to add another voicelink to this bookmark?" is output.	

Do you want to add another voicelink to this bookmark?	Click "Yes", "Is there any information here you would like to map?" is output. Click "No", "Saving Data..." is output.	

III. UDM Technical Specifications

This specification details the development of the functions and processes that are capable of accessing general browser-accessed content using UDM for retrieval from any device, such as a telephone, PC, or Internet-enabled PDA.

Requirements

UDM should be able to load any kind of data, such as networked data or content accessed from a browser. However, UDM cannot map the following types of data:

- Pages information that is dependent on JavaScript (VBScript) programs.
- SSL (Secure Sockets Layer) pages whose URL generally begin with "https://..."
- Any page that has an extension of .ps, .pdf, .cfm, or .exe.
- Pages with enabled flash objects.
- Video player pages.

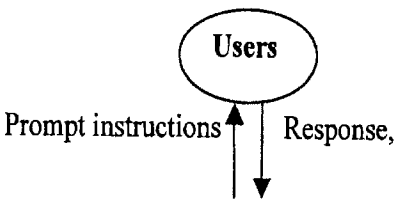
For those pages that UDM cannot load, UDM will display a standard error message to users, while the unloaded page and identified real error message are saved into a database table for TTtalk2 engineers who work with problem determination.

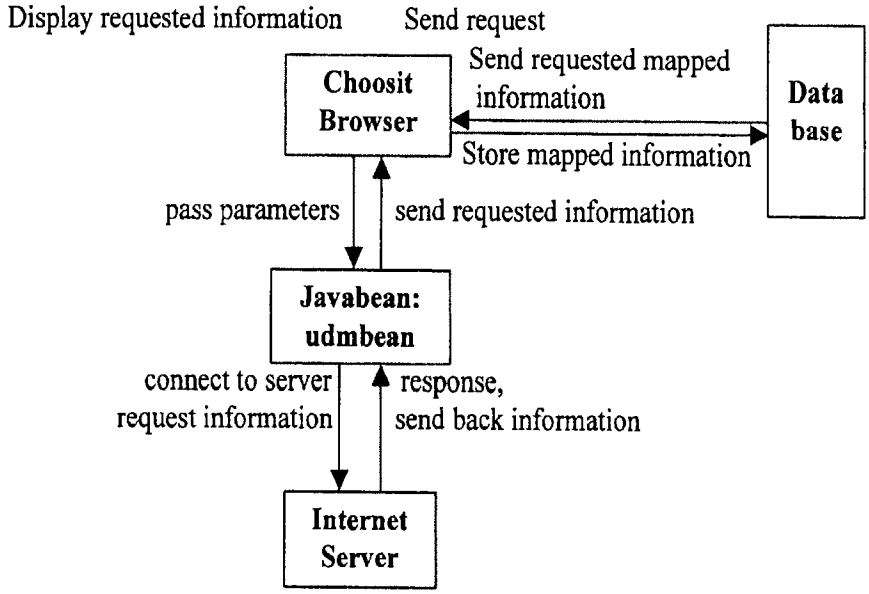
Design

See Figures 4 and 5 for a graphic depiction.

The user goes to the primary access portal and chooses the information s/he would like to map. Talk2 then goes to that site and pulls the requested content and returns the information to the user - Talk2 basically becomes a proxy. When it returns the text to the user Choosit also re-maps the content site and reroutes the location through the proxy rather than linking directly to the location. (See Figure 3 for a diagram of the UDM architecture.) The remapped location, such as a Web site, is still functional, but it has got a Talk2 wrapper around it. UDM actually parks the content and remaps it when Talk2 is acting as a proxy. Through this process, UDM is storing those offsets (collection of selected characters) locally on the user's box and doesn't actually save anything until the end. The system waits until the user gets to the end and it says "Do you want to add another voicelink to this bookmark," and the user says "no." Then, UDM submits all of the user's mapping to the server and saves it to the database. When the user is done, UDM sends it again to the Talk2 Web server and this time the Talk2 Web server writes it to work with database. This user-defined map is stored in the Talk2 database. (See Figure 6 for details about the database.) For example, a user would like to access Choosit-defined information by telephone. The user would dial into the telephone node and this database would get an identifier for this user and find the map associated with that person. This map is not just limited to the end-user, however, because there could also be a default map stored in the database regardless of who the user is.

Figure 3 **UDM Architecture**





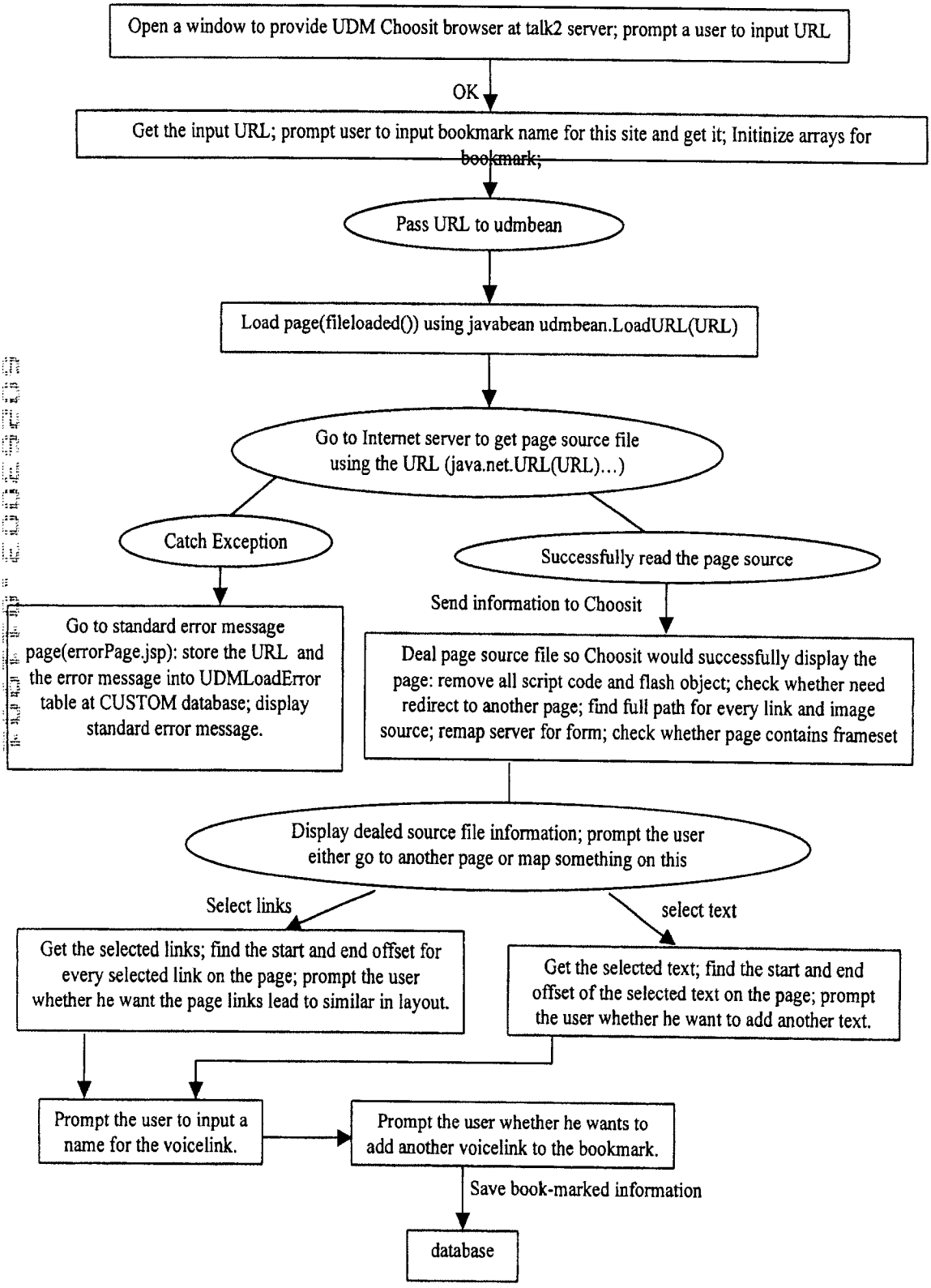
UDM actually saves the original content site as a collection of offsets in that site. UDM translates those offsets into the new page using what is called the **difference function. This difference function is how UDM picks up changes in the selected content.** For example, if a user would like to access Choosit Web content through a voice portal, the user calls into the telephony node, and the telephony node talks to the database and gets this original map of the content. The telephony node then talks to the site and gets the new page and determines the difference between the originally chosen content and the changes that may have been made to that URL and matched them up on a byte-by-byte basis.

If the chosen content location makes some changes to the selected data, the difference function allows the selected section to be dynamically updated. For example, if the function originally says, "I am a frog" and you want it to read the word "a frog." UDM creates a start pointer and an end pointer based on the start of the section and end of the section the user selects. The location changes the content so it says, "You are a funky frog." The difference function is going to find the best fit between these disparate phrases. In this case, the difference function will read through the text and will find a space followed by an "a." The difference function will determine there is probably a match between the spaces and words (in this case, "frog" would match). Based on the matches, the difference function will determine what characters were insertions and deletions. The difference function will then determine that based on the location of the deletion, the offset would move. In this example, UDM will start at the relocated offset and would stop, based on the original "end pointer". The system will say "a funky frog" instead of "a frog." With HTML code, this difference function works much better than this example demonstrates because HTML code actually has structure described in it and so the algorithm automatically selects the best way to match this content. If all of the text is the same, except slight changes, then UDM uses text. But if all other factors are different except for the structure, then UDM uses the structure.

Specification Detail

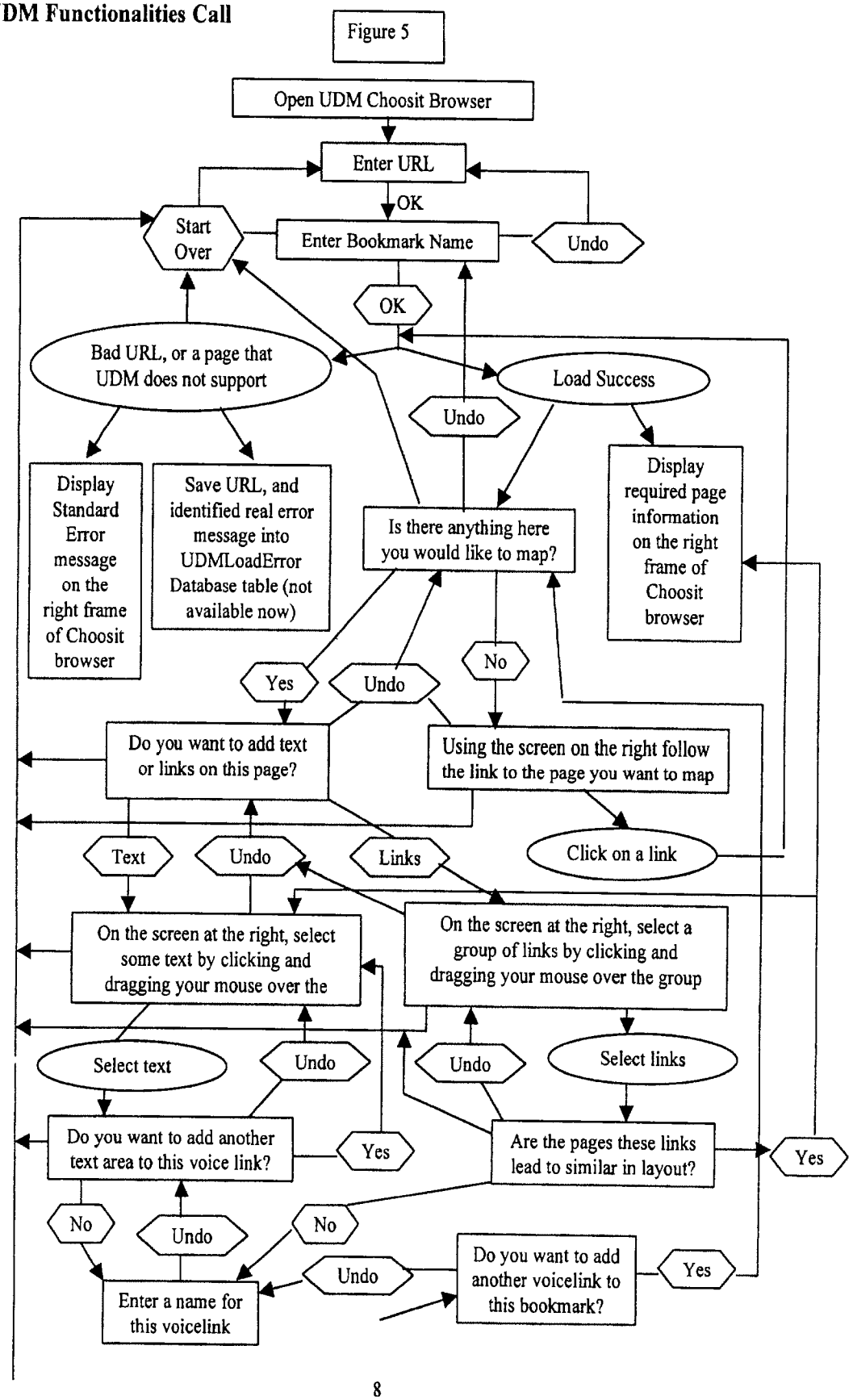
UDM High Level Design

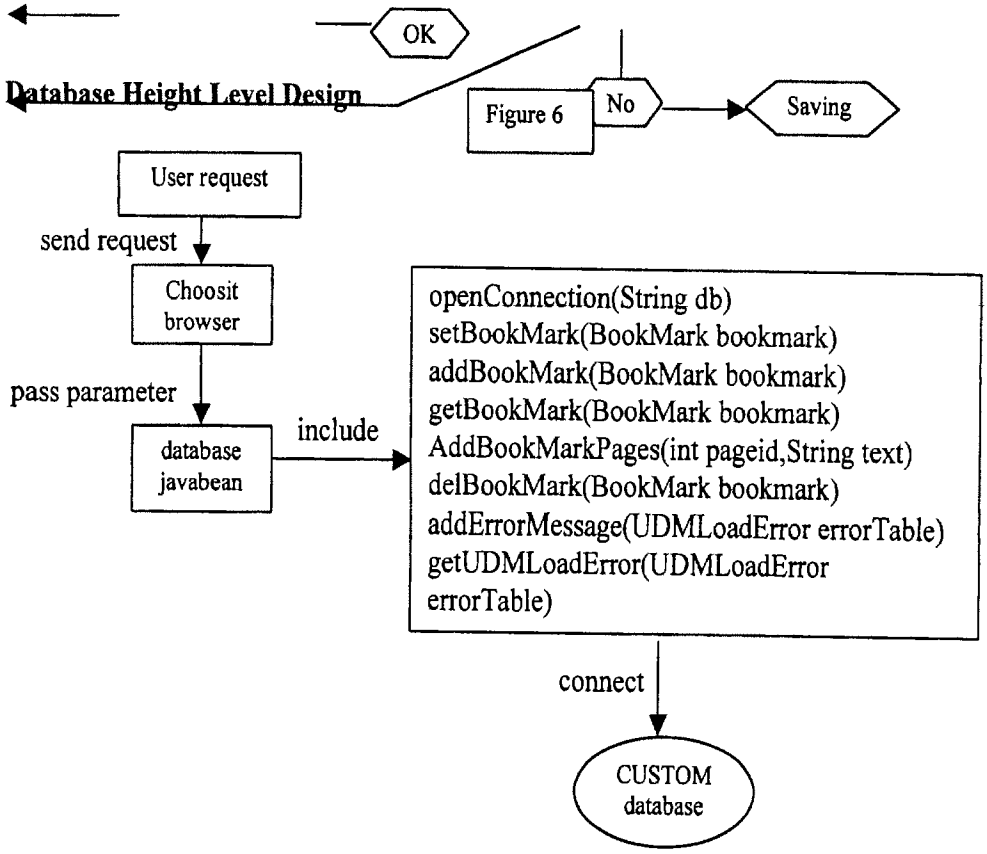
Figure 4



UDM Functionalities Call

Figure 5





Database schema

The following database tables are located in CUSTOM database.

BookMark table

Field Name	Type	Description
BookMarkID (key)	Number	ID assigned by UDM for this bookmark
AccountID	Number	This is the account ID for the owner of this bookmark
FolderType	Number	Option (value could be 1 to 9 that indicate which dial number is assigned to the bookmark)
BookMarkName	Text (50)	This will be the name for user to speak to access this bookmark (content)

BookMarkLink table

Field Name	Type	Description
BookMarkLinkID (key)	Number	ID assigned for each bookmark link
BookMarkID	Number	ID assigned for this bookmark

BookMarkLinkName	Text (50)	a name for this marked voicelink.
BookMarkLinkURL	Text (200)	URL of the web site which book marked some links
BookMarkLinkStartOfs	Number	Start offset of the book marked link on the web site
BookMarkLinkEndOfs	Number	End offset of the book marked link on the web site
BookMarkLinkOrigTextID	Number	ID assigned for the book marked link
BookMarkLinkTextOrigTextID	Number	ID assigned for the book marked text
BookMarkLinkIsLinkGroup	Number	Flag whether this bookmark is a group of links
BookMarkLinkIsDynamic	Number	Flag whether this book marked page is dynamic
BookMarkLinkPostData	Text (100)	Post data of this book marked link

BookMarkText table

Field Name	Type	Description
BookMarkLinkTextID (key)	Number	ID assigned for the mapped text
BookMarkLinkID	Number	ID assigned for the mapped link
BookMarkLinkTextStartOfs	Number	Start offset of the mapped text in the Web page
BookMarkLinkTextEndOfs	Number	End offset of the mapped text in the Web page

BookMarkPages table

Field Name	Type	Description
BookMarkPagesID (key)	Number	ID assigned for this web page.
BookMarkLargeText	Memo	Entire web page source file that the user maps some text or links from it.

UDMLoadError table

Field Name	Type	Description
UDMLoadErrorID (key)	Number	ID assigned for the error
URL	Text (250)	URL of the unloaded Web page
errorMessage	Text (150)	Identified real error (reason of why unable load the URL page)